



①9 BUNDESREPUBLIK
DEUTSCHLAND



DEUTSCHES
PATENT- UND
MARKENAMT

⑫ **Offenlegungsschrift**
⑩ **DE 101 12 409 A 1**

⑤① Int. Cl.⁷:
G 06 F 15/177

②① Aktenzeichen: 101 12 409.0
②② Anmeldetag: 13. 3. 2001
④③ Offenlegungstag: 19. 9. 2002

DE 101 12 409 A 1

⑦① Anmelder:
m-creations GmbH, 55116 Mainz, DE

⑦③ Vertreter:
Winter, M., Dipl.-Chem. Dr.rer.nat., Pat.-Anw., 71364
Winnenden

⑦② Erfinder:
Antrag auf Nichtnennung

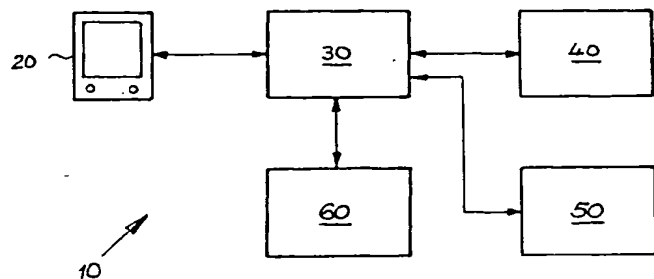
⑤⑥ Für die Beurteilung der Patentfähigkeit in Betracht
zu ziehende Druckschriften:

US 58 45 255 A
WO 99 63 473 A2

Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen

⑤④ Verfahren und System zur Datenverwaltung

⑤⑦ Die vorliegende Erfindung betrifft ein Verfahren zur Datenverwaltung, wobei Daten auf mindestens einem mit einer Software (Frontend) versehenen mobilen Rechner (20) eingegeben und bearbeitet werden, Daten auf mindestens einem Zentralrechner (40, 50) mittels einer Software (Backendsystem) gesammelt und gespeichert werden, wobei über mindestens einen lokalen Rechner (30) ein Datenaustausch zwischen dem mindestens einen mobilen Rechner (20) und dem mindestens einen Zentralrechner (40, 50) vorgenommen wird und die Software (Middleware) des lokalen Rechners (30) zumindest auch zur Zwischenübersetzung und Schnittstellenkontrolle dient. Erfindungsgemäß ist vorgesehen, dass auf dem mindestens einen Zentralrechner (40, 50) allgemeine, für jeden mobilen Rechner (20) zugängliche Daten gesammelt und gespeichert werden und dass mittels einer weiteren Software (Customiser) anwendungsspezifische, für einen oder mehrere mobile Rechner (20) individuelle Daten auf einem Rechner (20, 30, 60) gesammelt und gespeichert werden, wobei ein Datenaustausch zwischen dem mobilen Rechner (20) und dem Rechner (20, 30, 60) erfolgt, derart, dass die auf dem mobilen Rechner (20) vorhandene Software (Frontend) anwendungsspezifisch konfiguriert wird.



DE 101 12 409 A 1

Beschreibung

[0001] Die vorliegende Erfindung betrifft ein Verfahren zur Datenverwaltung nach dem Oberbegriff des Anspruchs 1 sowie ein System zur Datenverwaltung nach dem Oberbegriff des Anspruchs 9.

[0002] Der Gegenstand der Erfindung betrifft somit mobile Anwendungslösungen zur Datenverwaltung. Aufgrund der mobilen Natur einiger Berufsbilder besteht ein großer Bedarf an derartigen mobilen Anwendungslösungen, bspw. für sogenannte Handheld- oder Westentaschenrechner, die eine höchstmögliche Mobilität gewährleisten. Erwünscht sind hochmobile Anwendungen, die auf mobilen Kleinstgeräten laufen, Gruppenarbeit und Netzwerkfähigkeit sowie eine Anbindung an zentrale Datenbestände ermöglichen.

[0003] Ein besonderer Bedarf besteht zum Beispiel im Zusammenhang mit der Verwaltung von Patientendaten in einem Krankenhaus: Ein Arzt will am Bett seines Patienten während des Stationsbesuches eine Diagnose dokumentieren. Mit Hilfe einer mobilen Anwendung lädt er sich vorher die Patientendaten aus einem zentralen Informationssystem auf seinen mobilen Westentaschenrechner, dokumentiert einfach und schnell mit fertigen Textbausteinen und spielt diese wieder in das Krankenhaus-Dokumentationssystem ein. Dort können sie in der gewohnten Form weiterverarbeitet werden.

[0004] Ein derartiges gattungsgemäßes Verfahren bzw. System ist in der WO 99/41682 beschrieben. Das bekannte Verfahren bzw. System betrifft ausschließlich die Verwaltung von Patientendaten im Krankenhaus. Das bekannte System besteht im wesentlichen aus mindestens einem mobilen Rechner, einem lokalen Anwendungsrechner und mindestens einem Zentralrechner. Der lokale Anwendungsrechner vermittelt den Datenaustausch zwischen dem mobilen Rechner und dem Zentralrechner, bspw. einer Datenbank mit Patientendaten und stellt diese Patientendaten sowie Formblätter, Eingabemasken u. dgl. zur Eingabe von Patientendaten zur Verfügung. Somit kann der Arzt am Krankentisch Patientendaten abrufen, diese Daten aktualisieren bzw. neue Daten eingeben und auf dem lokalen Anwendungsrechner speichern. Die neuen Daten werden dann weiterverarbeitet, bspw. in eine auf einem Zentralrechner gespeicherte zentrale Patientendatei eingelesen oder neu formatiert, bspw. um die Daten in verschiedenen Datenbanken auf verschiedenen Zentralrechnern abzulegen. Die Datenübertragung und der Datenabgleich zwischen dem mobilen Rechner und dem lokalen Anwendungsrechner können synchron, bspw. über eine Telefon- oder Funkverbindung oder asynchron durch Andocken des mobilen Rechners an ein Terminal erfolgen.

[0005] Problematisch bei diesem Lösungsvorschlag ist, dass dieses Verfahren bzw. System nur sehr schwer an die individuellen Bedürfnisse einzelner Nutzer angepasst werden kann. Zwar können mit diesem System vorgegebene Listen gepflegt werden, aber grundsätzlich ist eine Änderung der auf dem mobilen Rechner ablaufenden Anwendung nur durch eine neue Programmversion möglich. Dies schränkt auch die Anwendung des bekannten Verfahrens bzw. Systems in weiteren Anwendungsbereichen wie Vertrieb, Lagerhaltung oder Service ein. Bei allen diesen Einsatzgebieten muss zur Umsetzung einer möglichst optimalen Lösung eine Verbindung der mobilen Einheit zu den Daten aus zentralen Systemen hergestellt und die Arbeit in der Gruppe ermöglicht werden. In den verschiedenen Einsatzgebieten bestehen unterschiedliche Anforderungen an sowohl die mobile Komponente als auch an die Verbindung zu den zentralen Datenbeständen. Alle bisher verfügbaren Lösungen berücksichtigen lediglich

die mobile Komponente als "Einzelpplatz", wobei nur ein Datenabgleich mit einem einzelnen Rechner möglich ist. Eine Anpassung der Einzelpplatzversion kann ausschließlich innerhalb der mobilen Anwendung selbst erfolgen, was zeitaufwendig und aufgrund der schlechten Eingabemöglichkeit bei mobilen Kleinstgeräten meist umständlich zu bewerkstelligen ist.

[0006] Die Aufgabe der vorliegenden Erfindung besteht somit darin, das bekannte Verfahren bzw. System derart weiter zu entwickeln, dass es in verschiedenen Einsatzgebieten an unterschiedliche Anforderungen der individuellen Anwender angepasst werden kann.

[0007] Damit die Anwendung individuell ausgestaltet werden kann, muss eine entsprechende, leicht zu bedienende und für alle Anwender zentral zu pflegende Anpassung ermöglicht werden.

[0008] Die Lösung besteht in einem Verfahren mit den Merkmalen des Anspruchs 1 sowie in einem Verfahren mit den Merkmalen des Anspruchs 9. Erfindungsgemäß ist also ein System vorgesehen, bei dem ein lokaler Rechner bzw. dessen Software (Middleware) die Kommunikation zwischen dem mobilen Rechner und dem Zentralrechner vermittelt, Daten abgleicht, konvertiert und/oder formatiert. Der lokale Rechner dient also vorwiegend als Zwischenübersetzer und Schnittstellenkontrollleur. Neben dem lokalen Rechner ist ein Zwischenrechner mit einer Anwendungssoftware (Customiser) vorgesehen, mit der die Software auf den mobilen Rechner an individuelle Bedürfnisse angepasst wird. Die zwischen dem zentralen Rechner und dem mobilen Rechner über den lokalen Rechner ausgetauschten Daten sind allgemeiner Natur und nicht individualisiert. Beispiele sind allgemeine Patientendaten in einer Krankenhausdatenbank, allgemeine Daten über Lagerbestände, allgemeine statistische Daten, allgemeine Verwaltungs- und Abrechnungsdaten. Die spezielle Anwendungssoftware auf dem Zwischenrechner (Customiser) liefert individualisierte Daten und individuelle Bearbeitungsvorlagen, bspw. anwendungsspezifische Formulare und Datenblätter zum Ausfüllen, standardisierte Formulareinträge, automatisierte Abfrageroutinen, Textbausteine, individuelle Daten, Listen und Protokolle, neue Menüstrukturen bzw. Anpassung bestehender Menüs. Damit können die vom Zentralrechner stammenden allgemeinen Daten individuell angepasst, ausgewertet und genutzt werden. Die Anwendungssoftware kann diese individualisierten Daten auch mehreren Nutzern mobiler Rechner mit gleichen Bedürfnissen (bspw. mehreren Ärzten der gleichen Abteilung, mehreren Außendienstmitarbeitern etc.) gleichzeitig zur Verfügung stellen, nachdem die Daten von einem Nutzer angepasst worden sind. Es findet also eine zentrale Anpassung der Software des mobilen Rechners statt. Die Anwendungssoftware erlaubt das einfache Gestalten persönlicher Programme und verarbeitet neue Daten und Programmelemente im Zusammenspiel mit dem Zentralrechner und/oder dem lokalen Rechner.

[0009] Die vorliegende Erfindung verfolgt somit einen generischen und modularen Ansatz, d. h. neben der Anbindung an die zentralen Datenbestände existiert eine separate Anwendungssoftware. Mit dieser Software werden typische "Arbeitsdaten", die häufig gebraucht werden, zentral zusammen gestellt und gepflegt. Diese Daten, aber auch Felder, Konfigurationen, Menüs, Menüstrukturen usw. werden dann auf den mobilen Rechner überspielt. Diese Individualkonfiguration kann durch Vermittlung eines lokalen Rechners zwischen mehreren Nutzern zusammenarbeiten. Nach der Individualkonfiguration ist also jeder berechnete Nutzer in der Lage, über den lokalen Rechner und dessen Software allgemeine Daten aus dem

Zentralrechner auf den mobilen Rechner zu laden und diese zu bearbeiten. Die bearbeiteten Daten werden rückgespeichert und über den lokalen Rechner dem Zentralrechner zur Verfügung gestellt. Damit kann jeder individuelle Nutzer die für seinen mobilen Rechner vorgesehenen Anwendungen einfach anpassen, z. B. Listen oder Listeneinträge bequem pflegen, so dass seinen Anforderungen individuell Sorge getragen werden kann.

[0010] Die grundlegende Idee der Lösung ist es, dem Nutzer ein einfaches Gestalten persönlicher Programme auf dem mobilen Gerät in einer Form zu ermöglichen, in der dieses Programm dann die neuen Elemente im Zusammenspiel mit z. B. einer zentralen Unternehmensdatenbank weiterhin verarbeiten kann.

[0011] Vorteilhafte Weiterbildungen ergeben sich aus den Unteransprüchen. Die Anbindung der mobilen Komponenten kann über ein Netzwerk erfolgen, so dass auch das Arbeiten in der Gruppe ermöglicht wird. Beim Abgleich des mobilen Systems können die Änderungen somit auf alle gewünschten Endgeräte übertragen werden. Dieser Vorgang funktioniert netzwerk, so dass die Anpassungen zentral vorgenommen werden können und anschließend an alle Benutzer automatisch verteilt werden.

[0012] Der Customiser kann auf einem Rechner, bspw. einem Server oder auf einem Desktoprechner für kleinere Änderungen, auch auf dem mobilen Gerät installiert sein. Darüber hinaus können Frontend und Middleware oder Frontend und Customiser oder Middleware und Customiser auf demselben Rechner, bspw. dem mobilen Rechner, dem Zwischenrechner oder auch dem lokalen Rechner installiert sein.

[0013] Die Middleware und der Customiser können inklusive der Schnittstellen in der Programmiersprache Java verwirklicht sein, da diese Programmiersprache aufgrund ihrer Datenbank Schnittstellen und ihrer Portabilität Vorteile in der Anwendung bietet.

[0014] Die Verbindung zwischen dem mobilen Rechner und dem lokalen Rechner kann asynchron, bspw. über ein Einsteckmodul oder synchron durch ein Netzwerk, auch eine kabellose Verbindung erfolgen. Die Kommunikation zwischen dem lokalen Rechner und dem Zentralrechner findet bevorzugt über standardisierte oder individuelle Schnittstellen statt.

[0015] Vorzugsweise ist der Zwischenrechner nur einem ausgewählten Benutzerkreis zugänglich, und der Customiser enthält Berechtigungskonzepte und -routinen zur Kontrolle der Zugangsberechtigung der jeweiligen Nutzer.

[0016] Ein Ausführungsbeispiel der vorliegenden Erfindung wird im folgenden anhand der beigelegten Zeichnungen näher erläutert. Es zeigen

[0017] Fig. 1 eine schematische Darstellung eines erfindungsgemäßen Systems zur Datenverwaltung;

[0018] Fig. 2 eine schematische Darstellung der Anwendung des erfindungsgemäßen Systems zur Verwaltung von Patientendaten;

[0019] Fig. 3 eine schematische Darstellung der Anwendung des erfindungsgemäßen Systems zur Qualitätskontrolle an Fertigungsstraten;

[0020] Fig. 4 eine schematische Darstellung der Anwendung des erfindungsgemäßen Systems für das Paybox-System (Zahlung per Mobiltelefon);

[0021] Fig. 5 eine Variante der Anwendung gemäß Fig. 4;

[0022] Fig. 6 bis 18 beispielhafte Darstellungen des Bildschirms eines mobilen Rechners für die verschiedenen An-

mobile Rechner wiederum kommuniziert mit einem lokalen Rechner 30, auf welchem eine weitere Softwarekomponente, im Folgenden Middleware genannt, installiert ist. Dieser lokale Rechner 30 stellt schließlich die Verbindung mit dem oder den Zentralrechnern 40, 50, her und vermittelt den Datenaustausch zwischen mobilem Rechner 20 und Zentralrechnern 40, 50. Auf dem oder den Zentralrechnern 40, 50 können verschiedene Systeme, bspw. Datenbanksysteme, installiert sein, die in Folgenden zusammenfassend als Backendsysteme bezeichnet werden. Zur Anpassung des Frontend an individuelle Anforderungen des jeweiligen Nutzers dient eine spezielle Anwendungskomponente, im Folgenden Customiser genannt, welche auf einem Zwischenrechner 60 installiert ist, der mit dem lokalen Rechner 30 kommuniziert.

[0024] Selbstverständlich können Customiser und Middleware auf einem Rechner installiert sein, ebenso wie bspw. Frontend und Customiser oder Frontend und Middleware.

Frontend

[0025] Das Frontend ist die Benutzerschnittstelle für den Anwender. Hiernit werden sowohl Daten angezeigt als auch durch den Nutzer manipuliert. Dazu werden verschiedene Masken und Listen zur Verfügung gestellt. Das Design der Benutzerschnittstelle ist speziell für die kleinen Endgeräte mit meist nur begrenztem Display optimiert.

[0026] Je nach Anforderung an die Mobilität und die Datenverfügbarkeit kann das Frontend synchron oder asynchron mit dem Backendsystem abgeglichen werden. Der synchrone Abgleich bedeutet, dass der mobile Rechner 20 für den Einsatz in einem kabellosen Netzwerk geeignet ist. Der Nutzer ist somit ständig online über den lokalen Rechner 30 mit dem oder den Zentralrechnern 40, 50 verbunden, so dass Änderungen sofort in die Middleware und/oder in das Backendsystem übernommen werden. Beim asynchronen Abgleich werden die Daten über eine stationäre Verbindung mit dem lokalen Rechner 30 abgeglichen. Verändert der Nutzer Daten auf dem mobilen Gerät, so werden diese Änderungen nach einem erneuten Abgleich in der Middleware und/oder im Backendsystem wirksam.

[0027] Die Frontendkomponente läuft generell auf allen mobilen Rechnern, wie Mobiltelefonen (bspw. mit UMTS-Standard), Handheldgeräten, Palmtops oder Laptops. Bevorzugt werden zur Zeit sogenannte Handheldgeräte (oder auch Westentaschenrechner, Personal Digital Assistants (PDA) etc.). Der meistverbreitete PDA wird von der Firma Palm™ hergestellt und mit dem Palm Operating System (Palm OS) betrieben. Weitere Palm OS Geräte werden u. a. von den Firmen Handspring und TRG hergestellt. Neben den Palm OS Geräten kann die Frontendhardware auf dem Betriebssystem WinCE der Firma Microsoft betrieben werden. Hier gibt es eine Vielzahl von Herstellern. Zukünftige Entwicklungen in diesem Bereich sind vor allem in der Migration von handelsüblichen Handys zu mobilen Kleincomputern zu erwarten.

[0028] Die Verbindung der mobilen Rechner 20 mit dem lokalen Rechner 30 kann auf verschiedene Weisen erfolgen. Sollen die mobilen Rechner 20 über ein kabelloses Netzwerk eingebunden werden, so sind z. B. DECT- oder Bluetoothnetze geeignet. Erfolgt der Datenabgleich über eine feste Station, so wird diese, bzw. der an die mobile Station an-

ponenten und Modulen. Jeder Nutzer kommuniziert mit dem System über einen mobilen Rechner 20 mittels einer Softwarekomponente, im Folgenden Frontend genannt. Der

lich. Ferner ist mit zukünftigen Entwicklungen zu rechnen wie z. B. Kombinationen (sog. Clones) aus den heutigen Handys und mobilen Computern.

[0029] Das Frontend kann vorzugsweise für mobile Kleincomputer, bspw. mit den Betriebssystemen PalmOS, Windows CE oder EPOC, verwirklicht werden. Durch die Verwendung einer portablen Programmiersprache, bspw. Java, ist es leicht möglich, andere existierende und neue Hardware-Geräte zu unterstützen. Das Frontend kann über eine eigens definierte Schnittstelle die durch den Customiser erzeugten Anwendungsvorgaben (welche Listen wohin, welche Menüs sollen welche Funktionen erfüllen etc.) in Form einer Anwendungsbeschreibung einlesen. Die Anwendungsbeschreibung entspricht daher in einem gewissen Sinne einer API (Application Programming Interface). Nach Einrichtung des Frontends werden über dieses die Daten, die von der Middleware geliefert wurden, verarbeitet, modifiziert und wieder an die Middleware zurückgereicht. In der Version für den lokalen Rechner 30 kann diese Funktionalität über Java Beans und in der Version für den mobilen Rechner 20 in einer direkten Form in eigenen Tabellen des mobilen Rechners realisiert werden.

Middleware

[0030] Damit das Frontend mit dem jeweils vorhandenen Backendsystem reibungslos kommunizieren kann, wird eine geeignete Middleware eingesetzt. Die Middleware wird auf einem lokalen Rechner 30 ausgeführt, der vorzugsweise je nach Ausprägung und Größe des Gesamtsystems ein handelsüblicher zentraler Server oder ein per Netzwerk angebundener Personal Computer (PC) sein kann. Betriebssystemseitig sind hier alle Lösungen möglich, vorzugsweise solche, auf denen die Programmiersprache Java zur Verfügung steht.

[0031] Die Kommunikation mit dem oder den Zentralrechnern 40, 50, also den Backendsystemen, erfolgt entweder auf der Basis von standardisierten Schnittstellen oder auf der Basis von speziell entwickelten Schnittstellenlösungen. In beiden Fällen erfolgt eine Übersetzung des benötigten Auszugs aus dem Datenmodell des Backendsystems in das Datenmodell des Frontendsystems und umgekehrt. Diese Übersetzung oder Formatierung wird von der Middleware vorgenommen.

[0032] Zusätzlich hat das Middleware-Programm auch vollständige Schnittstellen zu dem mobilen Rechner 20 (bei einem Palm™ wäre dies z. B. eine Erweiterung des sogenannten "HotSync"-Programmes). Damit ist gewährleistet, dass die Daten, die vom Customizer und von dem oder den Zentralrechnern 40, 50, also verschiedenen Backendsystemen (z. B. SAP) kommen, in geeigneter Form zusammengestellt werden und in der zusammengestellten Form dem mobilen Rechner 20 zur Verfügung gestellt werden.

[0033] Die Middleware kann in mehreren Versionen verwirklicht werden, bspw. je eine für den lokalen Rechner 30 und/oder Zwischenrechner 60 (z. Bsp. für die Betriebssysteme Linux und Windows) und für den mobilen Rechner 20. Bei letzterer handelt es sich bevorzugt um eine kleinere Middleware-Komponente, die bspw. auf einem Handheld läuft.

[0034] Die für den lokalen Rechner 30 bzw. den Zwischenrechner 60 vorgesehene Komponente der Middleware weist neben verschiedenen anderen Schnittstellen bspw. auch BAPI-Schnittstellen zu SAP R/3 in den Versionen 4.x der Module nm und IS-II auf. Hierzu wurden entlang der Programmierrichtlinien von SAP Business-Objekte kompiliert, die über Java Beans ansprechbar sind. In Richtung des Frontends (bspw. PDA der Fa. Palm) wird auch über Java eine Verbindungsschnittstelle angesprochen, die sich direkt in das HotSync-Protokoll des PDA einklinkt und damit die Synchronisation zum Frontend ermöglicht.

[0035] Innerhalb der Middleware läuft neben einem hochkomplexen Programmteil auch eine relationale Datenbank, die ein Ansprechen ihrer Tabellen über JDBC ermöglicht. In diese Datenbank werden von den Funktionen der Middleware die Daten aus verschiedenen Übertragungen vom Frontend auf Konsistenz geprüft und zusammengeführt. Zusätzlich spricht der Customiser über Java Beans diese Datenbank an, um hier eine Datenbank-unterstützte Katalogpflege zu ermöglichen. Auch die vom Benutzer auf dem mobilen Rechner 20 gewünschte Menü-Struktur wird in dieser Datenbank gespeichert. Aufgrund der nutzerspezifisch anzupassenden Menüstruktur wird die Datenbank direkt über JDBC angesprochen.

[0036] Die PDA-Komponente der Middleware ermöglicht unter der Verarbeitung der vorher durch den Customiser eingestellten Daten und Menüstrukturen die Verbindung zu einem Zentraldatenbank-System. Hierzu werden die Daten des Frontends mit den vom Customiser zur Verfügung gestellten Daten zusammengesetzt und per synchronem oder asynchronem Protokoll verarbeitet. Diese Version kann bspw. in einem Basic-Dialekt realisiert sein, da mit der aktuell verfügbaren Prozessortechnologie keine JVM (Java Virtual Machine, die Laufzeitgrundlage von Java auf einem Computer) mit akzeptabler Performance verfügbar ist.

Customiser

[0037] Je nach Einsatzgebiet und Aufgabenstellung soll das Frontend, also die konkrete Anwendung am mobilen Rechner 20, an die individuellen Vorgaben und Bedürfnisse jedes einzelnen Nutzers angepasst werden. Hierzu dient der Customiser. Durch ihn wird eine Anpassung der Frontendkomponente und damit des gesamten mobilen Systems von einem zentralen Punkt aus vorgenommen. Der Customiser läuft in der Regel auf einem PC oder einem Server (dies kann auch der PC oder Server sein, auf dem die Middleware installiert ist), der im Vergleich zu mobilen Endgeräten eine wesentlich effizientere Bearbeitung der Daten ermöglicht, kann aber auch, bei kleineren Anwendungen, auf dem mobilen Rechner 20 installiert sein.

[0038] Im Customiser werden die individuellen Daten bzw. die individuellen Listen ausgewählt und bearbeitet, die das Frontend auf dem mobilen Rechner 20 zur Verfügung stellen soll. Auch eine Anpassung der Menüs oder der Menüstruktur kann hier erfolgen. Nach einem Abgleich der Frontendkomponente mit der Middleware und/oder den Backendsystemen stehen die Änderungen den Endbenutzern zur Verfügung. Vorteilhaft ist in diesem Zusammenhang insbesondere, dass Listen etc. nicht von einzelnen Anwendern gepflegt werden müssen, sondern allen Nutzern zentral zur Verfügung gestellt werden, da sie von einem zentralen Administrator gepflegt werden. In der Regel hat nur ein ausgewählter Nutzerkreis Zugang zum Customiser. Dieser führt die entsprechenden Änderungen durch und macht sie somit allen mobil angeschlossenen Anwendern verfügbar. Die entsprechenden Berechtigungskonzepte sind ein Bestandteil des Programms.

[0039] Die Daten des Customisers werden in einer dafür vorgesehenen und leicht erweiterbaren Datenbankstruktur gehalten. Zur Neuanlage bzw. zur Änderung von individuellen Vorgaben für die mobile Anwendung steht eine entsprechende Benutzerschnittstelle zur Verfügung, die es dem Anwender ermöglicht, Daten schnell und bequem zu ändern oder anzufügen.

[0040] Auch der Customiser kann bspw. in einer Version für den Zwischenrechner 60 und/oder lokalen Rechner 20 oder in einer Version für den mobilen Rechner 20 zur Verfügung gestellt werden.

[0041] Die Version für den Zwischenrechner **60** bzw. den lokalen Rechner **30** kann ebenfalls in Java realisiert sein. Der Customiser übersetzt eine benutzerfreundliche Oberfläche zum Beispiel mit Hilfe von Clickboxen, geschrieben mit den Java Swing-Bibliotheken, in eine Struktur von Textbausteinen, Listen und Menüs, die beim nächsten Synchronisationsbefehl auf dem mobilen Rechner **20** sichtbar werden. Um die gleichzeitige Unterstützung mehrerer mobiler Rechner **20** zu gewährleisten, wird die Ablage der vom Customiser zur Verfügung gestellten Daten in einer eigenen Datenbank sichergestellt. Notwendige neue Tabellen werden hier über JDBC erstellt, ansonsten über die generierten Beans angesprochen. Diese Datenbankinhalte werden dann durch ein Funktionsset im Customiser, das die Textbausteine und Menüstrukturen ausliest, für das Frontend des mobilen Rechners **20** umgesetzt, wobei vorher eine Konsistenzprüfung auf die Datenstrukturen des lokalen Rechners **30** bzw. der Middleware erfolgt. Bei der Umsetzung auf mehrere mobile Rechner **20** wird jeweils dieses Funktionsset angesprochen, das dann die für den entsprechenden mobilen Rechner **20** gewünschte Listen-, Textbaustein- und Menüstrukturen übersetzt und installiert.

[0042] Die kleinere Version des Customisers für den mobilen Rechner **20**, insbesondere einen PDA, unterstützt vorzugsweise keine Definition von Menüstrukturen, sondern nur die Definierbarkeit von Listen und Textbausteinen. Sie ist in dem oben genannten Basic-Dialekt verwirklicht, ansonsten strukturell der größeren Version des Customiser ähnlich, wenn auch sehr vereinfacht.

Backendsysteme

[0043] Als Backendsysteme kommen verschiedene zentrale Datenhaltungssysteme in Frage. Dazu gehören u. a. Krankenhausinformationssysteme (z. B. SAP R/3 mit den Modulen IS-II und IS-II*^{nied}), Enterprise-Ressource-Planning (ERP) Systeme, Vertriebsinformationssysteme und Customer-Relationship-Management (CRM) Systeme. Die Anbindung des lokalen Rechners **30** zu den Zentralrechnern **40**, **50** bzw. den Backendsystemen erfolgt vorzugsweise über ein Netzwerk. Der Datenaustausch erfolgt entweder über eine Standardschnittstelle oder eine speziell erstellte Kommunikationsssoftware.

[0044] Das erfindungsgemäße Verfahren kann wie folgt durchgeführt werden: Zunächst legitimiert sich der Nutzer an dem jeweiligen mobilen Rechner **20** mit Benutzername und Kennwort. Bei erfolgreicher Anmeldung erhält er nach einem Abgleich mit der Middleware des lokalen Rechners **30** die aktuell verfügbare Information aus dem zentralen Datenbestand des Backendsystems (z. B. Kundendaten) sowie aus dem Customiser (z. B. neue Vorgabetabellen und Vorlagen). Die Informationen werden dem Anwender in Listenform oder in Form von Auswahlfeldern zur Verfügung gestellt. Suchfunktionen über den Datenbestand oder über einzelne Listen stehen ebenso zur Verfügung. Zur Manipulation oder Neueingabe von Daten stehen dem Anwender verschiedene Werkzeuge, bspw. Auswahllisten, vordefinierte Textbausteine, Ja-Nein-Felder und die Eingabe von freiem Text, zur Verfügung. Nach erfolgten Änderungen oder Neueingaben wird das Frontend wiederum mit dem Backendsystem abgeglichen, so dass die zentral gehaltenen Daten wieder auf den aktuell gültigen Stand gebracht werden.

[0045] Will ein Anwender einen Abgleich der Daten des Frontend mit dem Backendsystem durchführen, wird zunächst eine Verbindung zwischen dem mobilen Rechner **20** und dem lokalen Rechner **30** hergestellt. Dort wird dann die Anfrage bearbeitet. Sollen Daten aus dem Backendsystem angezeigt oder geändert werden, stellt der

lokale Rechner **30** eine Verbindung zu dem oder den Zentralrechnern **40**, **50**, also zum Backendsystem her, vorzugsweise über ein Netzwerk. Anschließend werden die angeforderten Daten aus dem Backendsystem geladen bzw. geänderte Daten in das Backendsystem zurückgeschrieben. Die Middleware sorgt auch für die Verarbeitung eines Abgleichs mit den Daten des Customiser.

[0046] Auf dem mobilen Gerät können die Daten dann mit dem Frontend vor Ort, also z. B. am Krankenbett oder beim Kunden, bearbeitet werden, es können Dokumentationen erstellt, Kalkulationen durchgeführt oder Anfragen ausgeführt werden. Die so bearbeiteten Daten werden dann über die Middleware zwischenformatiert und zurück auf den oder die Zentralrechner **40**, **50**, bspw. eine eigene Datenbank oder eine Zentraldatenbank (z. B. SAP) übertragen.

[0047] Im Folgenden sollen verschiedene Anwendungsbeispiele des erfindungsgemäßen Systems bzw. Verfahrens beschrieben werden.

1. Verwaltung von Patientendaten (vgl. Fig. 2, Fig. 6 bis 14)

[0048] Die Funktion des Customisers für diese Anwendung lässt sich wie folgt beschreiben: Jede Fachrichtung erbringt verschiedene Leistungen und muss unterschiedliche Diagnosen stellen. Im Customiser werden die für die jeweilige Abteilung relevanten Diagnosen, Medikamente und Leistungen in Form von Listen erstellt. Des weiteren können neue Listen zur Speicherung von Textbausteinen für z. B. Arztbriefe, Stationslisten oder Telefonnummern erstellt werden, aber auch Listen mit den für die Anwender häufigsten ICD10-Verschlüsselungscodes und OPS zur automatischen Verschlüsselung von Diagnosen und Leistungen direkt auf dem mobilen Rechner **20**. Dies erfolgt im Normalfall lediglich auf dem lokalen Rechner **30**, bspw. einem Abteilungsserver. Nach einem Abgleich des mobilen Rechners **20** mit dem lokalen Rechner **30** erhält jeder das System nutzende Arzt automatisch die aktuelle Fassung der Listen und Textbausteine.

[0049] Vor dem Patientenbesuch gibt der Arzt (bei einem neu aufgenommenen Patienten) die relevanten Patientendaten in ein Formular ein (vgl. Fig. 6), welches vom Customiser abteilungsspezifisch zur Verfügung gestellt wird. Andernfalls gleicht der Arzt (z. B. asynchron) auf seinem mobilen Rechner **20** den für ihn relevanten Patientendatenbestand ab (vgl. Fig. 7, 8). Damit stehen ihm die aktuellen Patientendaten zur Verfügung. Im Laufe des Patientenbesuchs erfasst er z. B. Diagnose- und Leistungsdaten (vgl. Fig. 9) und fügt Anordnungen ein. Dazu stellt ihm der Customiser z. B. eine Liste der möglichen bzw. in der entsprechenden Fachrichtung häufigsten Diagnosen (vgl. Fig. 10, 11) und Medikamente (Fig. 12, 13, bspw. die gängigsten Analgetika) sowie ein auf seine Abteilung angepasstes Set an Textbausteinen und weiteren Listen zur Verfügung. Der Arzt braucht also die einzelnen Positionen nicht mehr von Hand in den mobilen Rechner **20** einzugeben, sondern es genügt ein einfaches Antippen einer Position in einer Liste. Nach erfolgtem Patientenbesuch führt er wiederum einen Abgleich durch (vgl. Fig. 14). Die neuen Daten sind damit zur weiteren Bearbeitung im oder in den Zentralrechnern **40**, bspw. im Krankenhausinformationssystem, abgelegt.

[0050] Beim ersten Abgleich werden vom lokalen Rechner **30** zunächst Daten vom mobilen Rechner **20** bzw. vom Frontend angefordert. Die Middleware überprüft die Legitimation des Arztes und stellt die Verbindung zum Zentralrechner (Zentralrechner des Krankenhausinformationssystems (KIS) her, um die Daten (z. B. Patientendaten) an den mobilen Rechner **20** bzw. das Frontend zu übertragen. Die Verbindung zum KIS wird z. B. über eine standardisierte HL7-

Schnittstelle hergestellt. Nach dem Patientenbesuch des Arztes mit z. B. Dokumentationen oder Laboranforderungen auf dem mobilen Rechner 20 bzw. Frontend erfolgt ein erneuter Abgleich der Daten des mobilen Rechners 20 in Richtung zum KIS über den lokalen Rechner 20 bzw. die Middleware. Sollen neue, individuell angepasste Listen (z. B. Fachrichtungsspezifische Diagnosen) oder Anwendungsdaten auf den mobilen Rechner 20 bzw. das Frontend eingespielt werden, erfolgt über die Middleware ein Abgleich mit den Daten des Customisers.

2. Kontrolle von Fertigungsstraßen (Fig. 3)

[0051] Ertrag einer Fertigungsstraße 80, bspw. in der Automobilproduktion, sind an einzelnen Fertigungsetappen 80a, 80b, 80c, Qualitätskontrollen eingerichtet. Jeder Kontrollpunkt ist mit einem mobilen Rechner 20 ausgerüstet, welcher über einen eigenen lokalen Rechner und Customiser mit spezifischen Daten (Tabellen, Listen etc.) für die vorgegebene Qualitätskontrolle versorgt wird. Selbstverständlich können die einzelnen lokalen Rechner 30 und/oder Customiser auf einem gemeinsamen Rechner, bspw. einem Server, abgelegt sein. Jeder Nutzer an jedem Kontrollpunkt gibt für jedes kontrollierte Produkt die entsprechenden Daten in den mobilen Rechner ein bzw. lädt die vom jeweiligen Customiser vorgegebenen Formulare aus. Die Daten werden über den oder die lokalen Rechner 30 zu einem Zentralrechner 50 gesendet, welcher die Daten für jedes Produkt zusammenführt und auswertet. Der lokale Rechner 30 oder der Zentralrechner 50 liefern auch für jedes Produkt am Ende der Fertigungsstraße einen Qualitätsbericht mit einem aufsummierten Fehlerprozent, d. h. den jeweiligen, an den einzelnen Kontrollpunkten gemachten Fehlermeldungen, der am Ende der Fertigungsstraße ausgedruckt wird.

3. Autendienst (Fig. 1)

[0052] Mit dem erfindungsgemäßen System bzw. Verfahren können Außendienstmitarbeiter mit dem mobilen Rechner 20 bspw. Lagerbestände abfragen oder Informationen zu Angeboten und Konditionen einholen und Bestellungen aufgeben. Je nach Aufgabengebiet liefert der Customiser bspw. Antworten zu FAQs der Kunden, automatisierte Lagerbestandsabfragen, wobei die am häufigsten gefragten Produkte zuerst gelistet werden, Textbausteine für häufige Abfragen am Zentralrechner oder häufige Bestellungen, etc. Das erfindungsgemäße System kann hier also bspw. als Servicesystem, CRM-Modul (customer relationship management) oder als mobiles Bestell- bzw. Kaufsystem (Sales-System) dienen.

4. Bezahlen von Taxifahrern mit dem Paybox-System (Fig. 4)

[0053] In diesem Fall ist der Taxifahrer der Nutzer des mobilen Rechners 20, der gleichzeitig als Zwischenrechner 60 dient, weil der Customiser im mobilen Rechner integriert ist. Der Customiser stellt die individuellen Daten des Taxifahrers bzw. Taxifahrers zur Verfügung, während das Frontend den Rechnungsbetrag ermittelt und bspw. über ein GSM-Modul die Telefonverbindung zum lokalen Rechner 30 herstellt. Der lokale Rechner 30 wiederum vermittelt zwischen den verschiedenen Frontends einzelner mobiler Rechner 20 (z. B. Bus, Mobiltelefon, PDA, etc.) und dem Zentralrechner 40. Der lokale Rechner 30 fungiert als Zentralrechner 40. Dann kann der Rückruf auf das Mobiltelefon des Fahrgastes erfolgen, mit dem dieser die Zahlung bestätigt.

5. Bezahlen von Lieferdiensten mit Paybox am Beispiel eines Pizza-Service (Fig. 5, Fig. 15 bis 18)

[0054] Hier ist der Bote des Lieferdienstes der Nutzer des mobilen Rechners 20. Über den Customiser wird dem mobilen Rechner die Speise- und Getränkekarte des Lieferdienstes samt Preisen mittels Textbausteinen zur Verfügung gestellt. Das Frontend enthält ein Programm zur Ermittlung des Rechnungsbetrages sowie die Middleware zur Herstellung der Telefonverbindung mit dem Paybox-Zentralrechner 40. Zur Bezahlung des Rechnungsbetrages tippt der Bote die bestellten Speisen im mobilen Rechner an (Fig. 15, 16). Das Frontend ermittelt den Rechnungsbetrag (vgl. Fig. 17), nimmt Verbindung mit dem Paybox-Zentralrechner auf (vgl. Fig. 18) und übermittelt die Zahlungsdaten des Kunden. Dann erfolgt wieder der Rückruf vom Zentralrechner 40 zum Mobiltelefon des Kunden zur Abbuchungsbestätigung.

Patentsprüche

1. Verfahren zur Datenverwaltung, wobei Daten auf mindestens einem mit einer Software (Frontend) versehenen mobilen Rechner (20) eingegeben und bearbeitet werden, Daten auf mindestens einem Zentralrechner (40, 50) mittels einer Software (Backendsystem) gesammelt und gespeichert werden, wobei über mindestens einen lokalen Rechner (30) ein Datenaustausch zwischen dem mindestens einen mobilen Rechner (20) und dem mindestens einen Zentralrechner (40, 50) vorgenommen wird und die Software (Middleware) des lokalen Rechners (30) zumindest auch zur Zwischenübersetzung und Schnittstellenkontrolle dient, dadurch gekennzeichnet, dass auf dem mindestens einen Zentralrechner (40, 50) allgemeine, für jeden mobilen Rechner (20) zugängliche Daten gesammelt und gespeichert werden, und dass mittels einer weiteren Software (Customiser) anwendungsspezifische, für einen oder mehrere mobile Rechner (20) individuelle Daten auf einem Rechner (20, 30, 60) gesammelt und gespeichert werden, wobei ein Datenaustausch zwischen dem mobilen Rechner (20) und dem Rechner (20, 30, 60) erfolgt, derart, dass die auf dem mobilen Rechner (20) vorhandene Software (Frontend) anwendungsspezifisch konfiguriert wird.
2. Verfahren nach Anspruch 1, dadurch gekennzeichnet, dass der Datenaustausch zwischen dem mobilen Rechner (20) und dem Rechner (20, 30, 60) über die Software (Middleware) des lokalen Rechners (30) erfolgt.
3. Verfahren nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, dass die weitere Software (Customiser) auf einem separaten Zwischenrechner (60) gespeichert wird.
4. Verfahren nach einem der Ansprüche 1 und 2, dadurch gekennzeichnet, dass die weitere Software (Customiser) auf dem lokalen Rechner (30) gespeichert wird.
5. Verfahren nach einem der Ansprüche 1 und 2, dadurch gekennzeichnet, dass die weitere Software (Customiser) auf dem mobilen Rechner (20) gespeichert wird.
6. Verfahren nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, dass Frontend und Customiser oder Frontend und Middleware oder Customi-

ser und Middleware auf demselben Rechner gespeichert werden.

7. Verfahren nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, dass die weitere Software (Customiser) mit mindestens einem Berechtigungskonzept zur Zugangskontrolle versehen wird. 5

8. Verfahren nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, dass der mindestens eine mobile Rechner (20) über ein Netzwerk an den mindestens einen lokalen Rechner (30) und/oder den mindestens einen Zwischenrechner (60) und oder den Zentralrechner (40, 50) angebunden wird. 10

9. System zur Datenverwaltung mit mindestens einem mit einer Software (Frontend) versehenen mobilen Rechner (20) zur Eingabe und Speicherung von Daten. 15

mindestens einem Zentralrechner (40, 50) mit einer Software (Backendsystem) zum Sammeln und zur Speicherung von Daten.

mindestens einen lokalen Rechner (30) zum Datenaustausch zwischen dem mindestens einen mobilen Rechner (20) und dem mindestens einen Zentralrechner (40, 50), wobei die Software (Middleware) des lokalen Rechners (30) zumindest auch zur Zwischenübersetzung und Schnittstellenkontrolle dient. 20

dadurch gekennzeichnet, dass eine weitere Software (Customiser) vorgesehen ist, mit der anwendungsspezifische, für einen oder mehrere mobile Rechner (20) individuelle Daten auf einem Rechner (20, 30, 60) gesammelt und gespeichert werden können. 25

wobei ein Datenaustausch zwischen dem mobilen Rechner (20) und dem Rechner (20, 30, 60) erfolgt, derart, dass die auf dem mobilen Rechner (20) vorhandene Software (Frontend) anwendungsspezifisch konfiguriert ist. 30

10. System nach Anspruch 9, dadurch gekennzeichnet, dass ein Zwischenrechner (60) vorgesehen ist, auf dem die weitere Software (Customiser) gespeichert ist.

11. System nach einem der Ansprüche 9 und 10, dadurch gekennzeichnet, dass Frontend und Customiser oder Frontend und Middleware oder Customiser und Middleware auf demselben Rechner gespeichert sind. 35

12. System nach einem der Ansprüche 9 bis 11, dadurch gekennzeichnet, dass die weitere Software (Customiser) mit mindestens einem Berechtigungskonzept zur Zugangskontrolle versehen ist. 40

13. System nach einem der Ansprüche 9 bis 12, dadurch gekennzeichnet, dass der mindestens eine mobile Rechner (20) über ein Netzwerk an den mindestens einen lokalen Rechner (30) und/oder den mindestens einen Zwischenrechner (60) und oder den Zentralrechner (40, 50) angebunden ist. 45

14. System nach einem der Ansprüche 9 bis 13, dadurch gekennzeichnet, dass Frontend und/oder Middleware und/oder Customiser in der Programmiersprache Java realisiert sind. 50

15. Verwendung des Verfahrens nach einem der Ansprüche 1 bis 8 und/oder des Systems nach einem der Ansprüche 9 bis 14 zur Verwaltung von Patientendaten in einem Krankenhaus. 55

16. Verwendung des Verfahrens nach einem der Ansprüche 1 bis 8 und/oder des Systems nach einem der Ansprüche 9 bis 14 für den Zahlungsverkehr mit der

stelldaten.

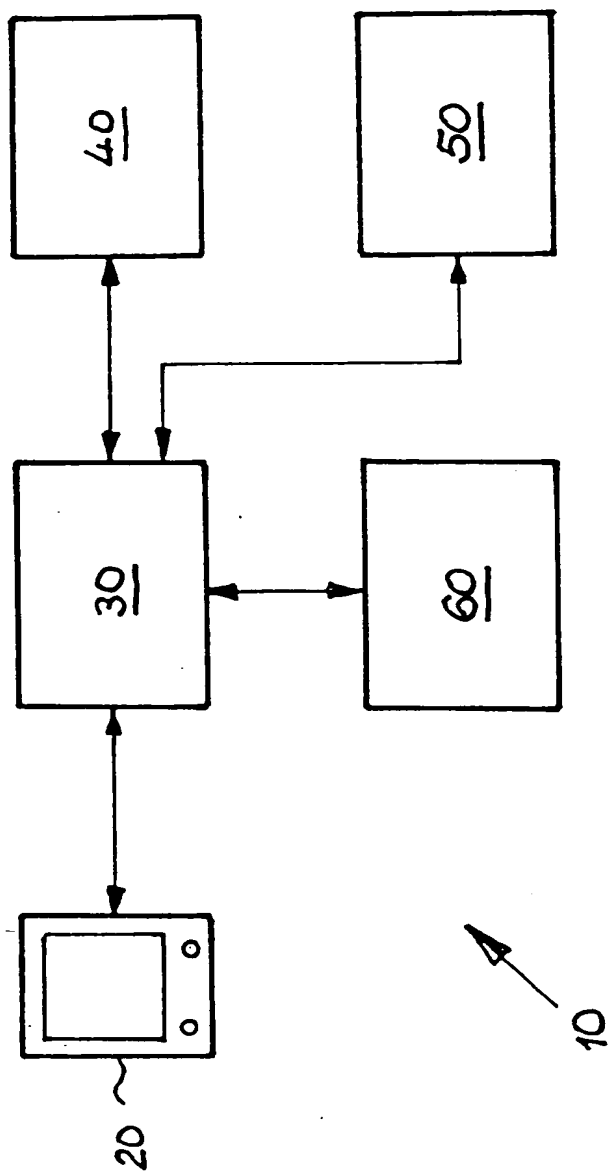
18. Verwendung des Verfahrens nach einem der Ansprüche 1 bis 8 und/oder des Systems nach einem der Ansprüche 9 bis 14 zur Qualitätskontrolle in der Produktion, insbesondere an Fertigungsstraßen.

19. Verwendung des Verfahrens nach einem der Ansprüche 1 bis 8 und/oder des Systems nach einem der Ansprüche 9 bis 14 als Kaufsystem (Sales-System)-, CRM- und/oder Servicesystem.

Hierzu 18 Seite(n) Zeichnungen

- Leerseite -

Fig. 1



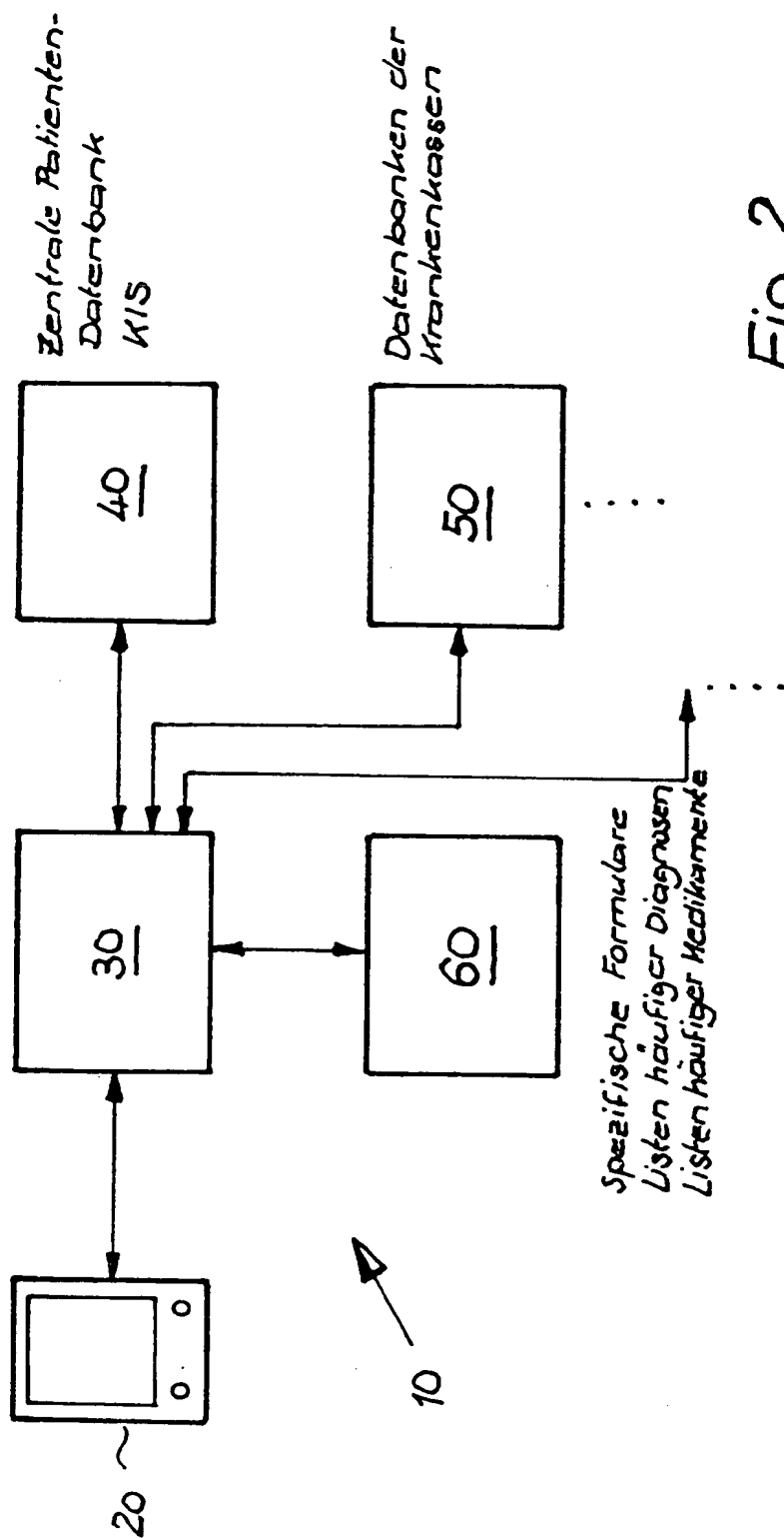


Fig. 2

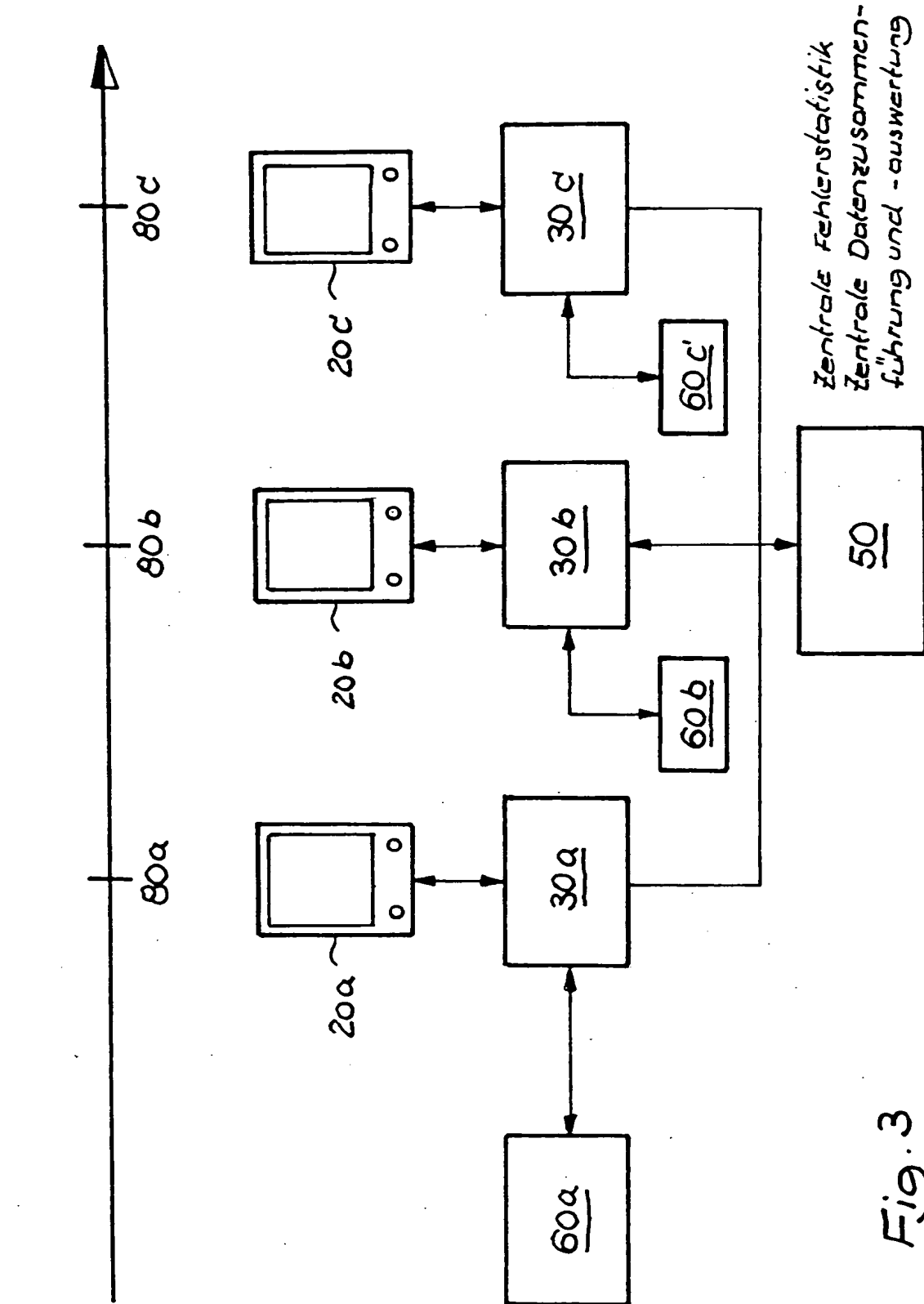


Fig. 3

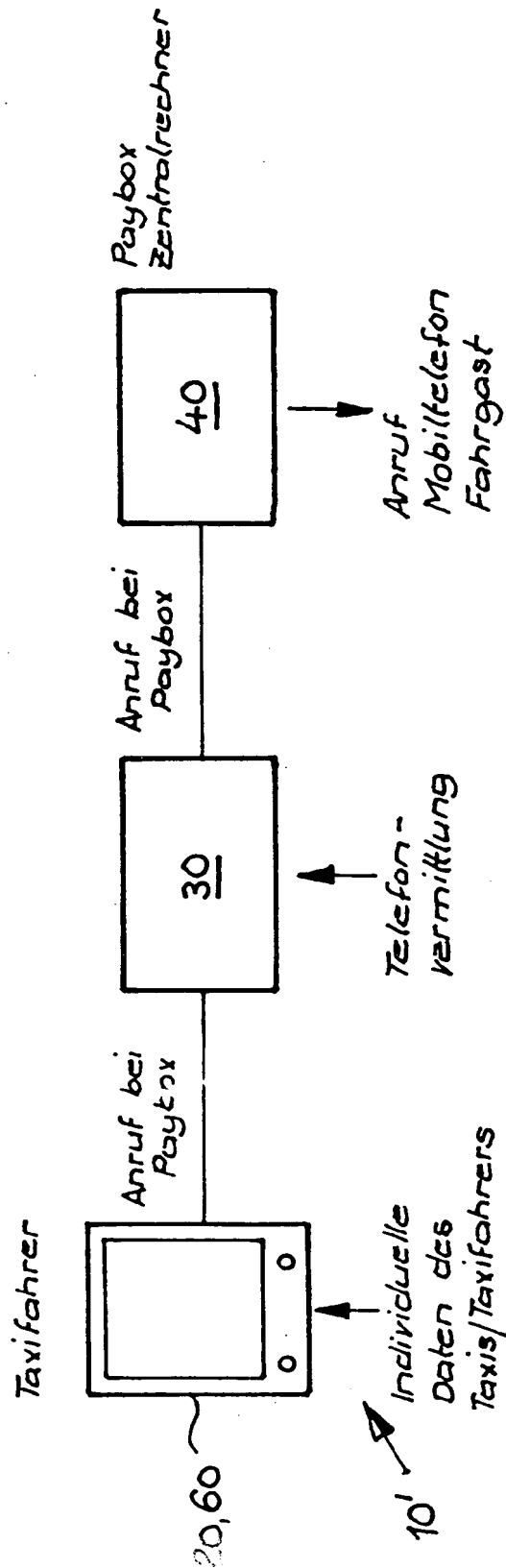


Fig. 4

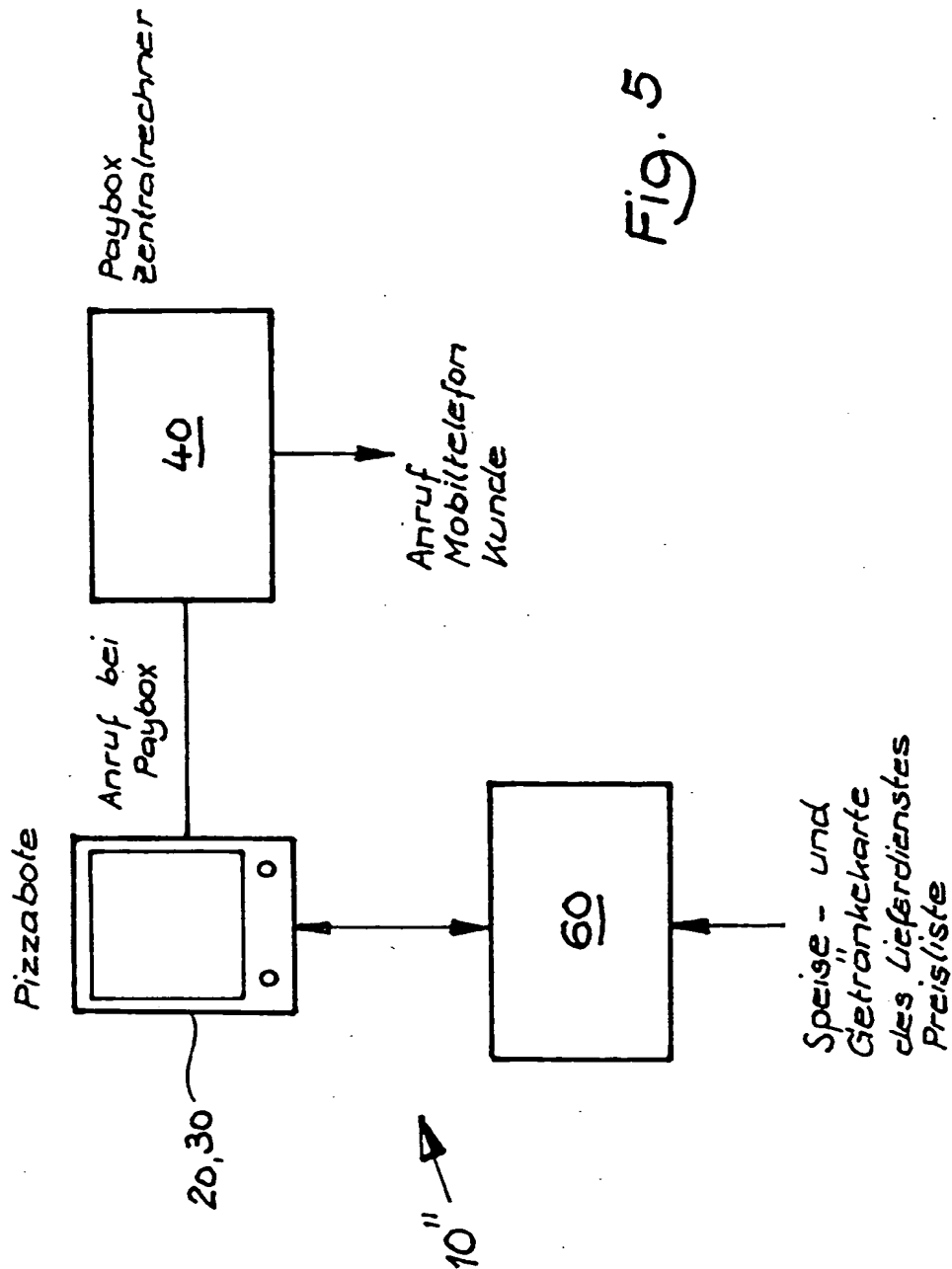
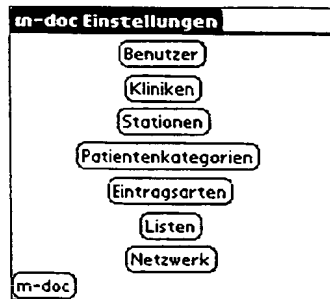
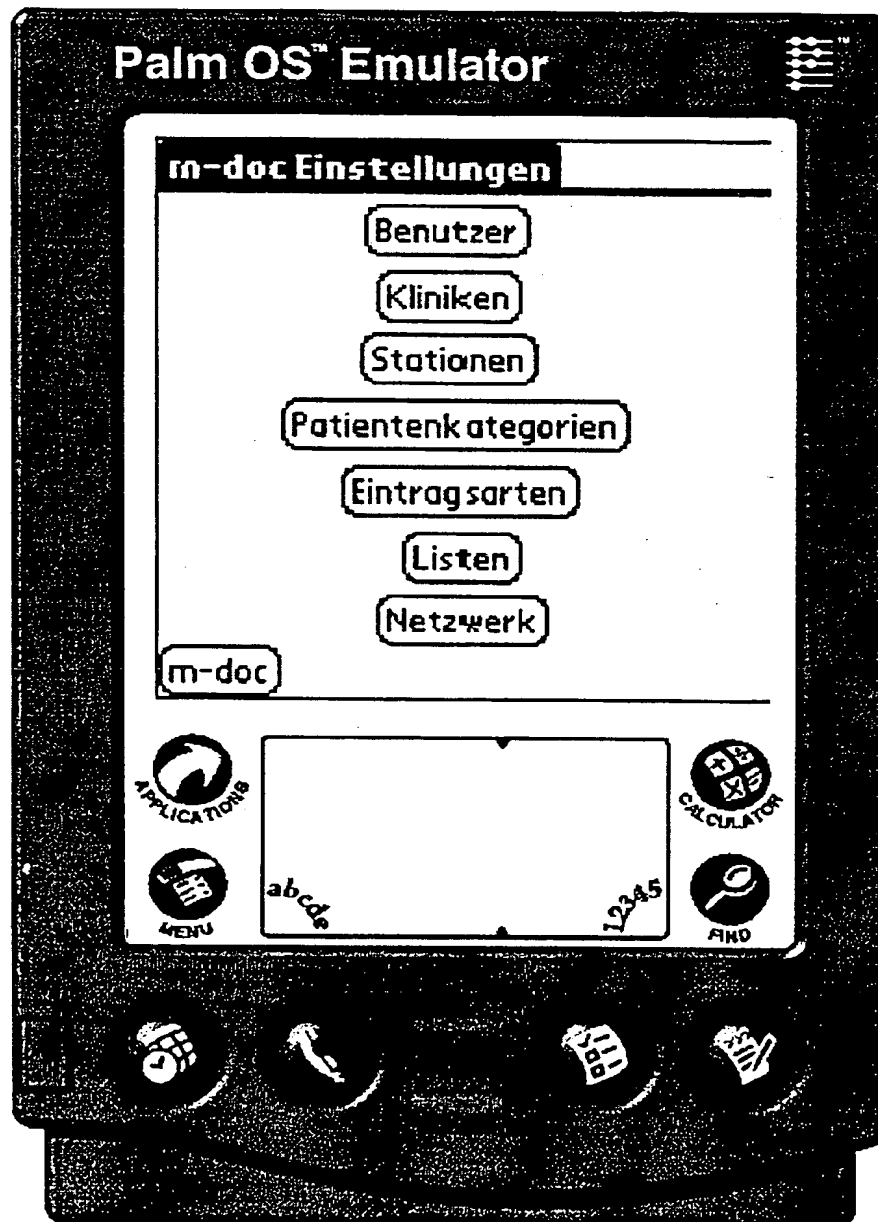


Fig. 5

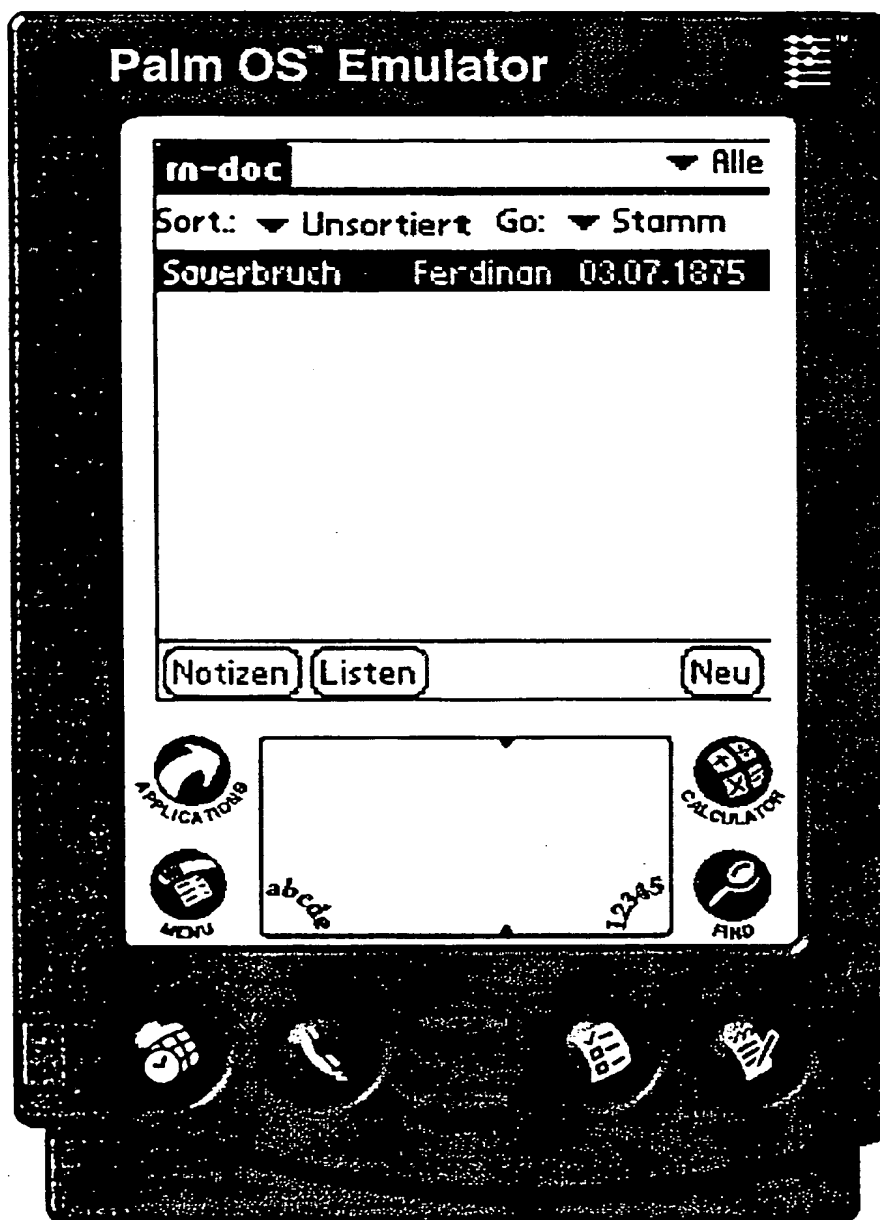


Figur 6



m-doc	▼ Alle
Sort: ▼ Unsortiert	Go: ▼ Stamm
Sauerbruch Ferdinand 03.07.1875	
Notizen	Listen
Neu	

Figur 7



BEST AVAILABLE COPY

ZEICHNUNGEN SEITE 8

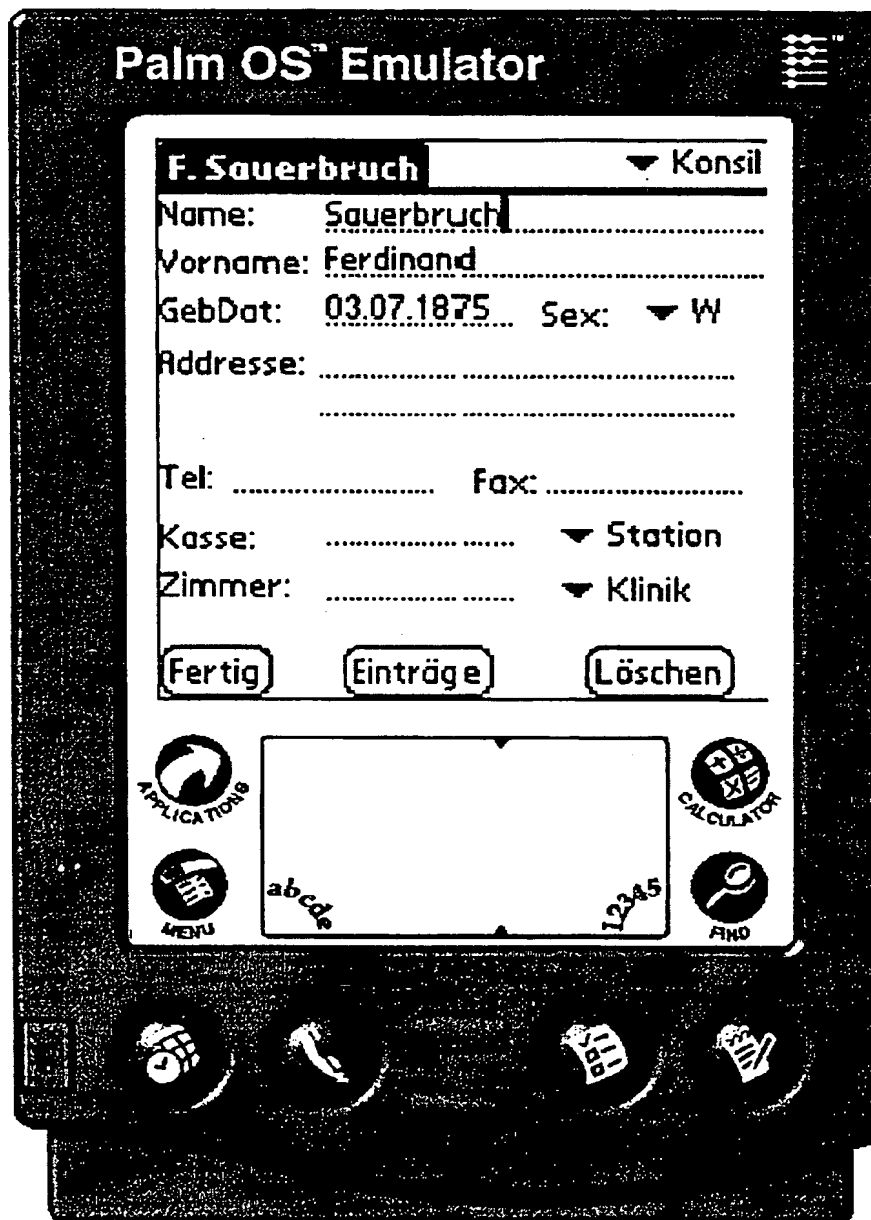
Numm
Int. Cl./:
Offenlegungstag:

DE 101 12 409 A1
G 06 F 15/177
19. September 2002

F. Sauerbruch ▼ Konsil

Name: Sauerbruch
Vorname: Ferdinand
GebDat: 03.07.1875 Sex: ▼ W
Adresse:
Tel: Fax:
Kasse: ▼ Station
Zimmer: ▼ Klinik
Fertig Einträge Löschen

Figur 8



Sauerbruch	Stat. Aufnahme
Keine Einträge	Diagnose
	Medikamente
	Allergie
	Anamnese
	F. Anamnese
	N. U'suchung
	OP-Planung
	Operation
	Visite
	Verband
	Konsil
	Lab. U'suchung
	Rö. U'suchung

(Fertig) (Drucken)

Figur 9



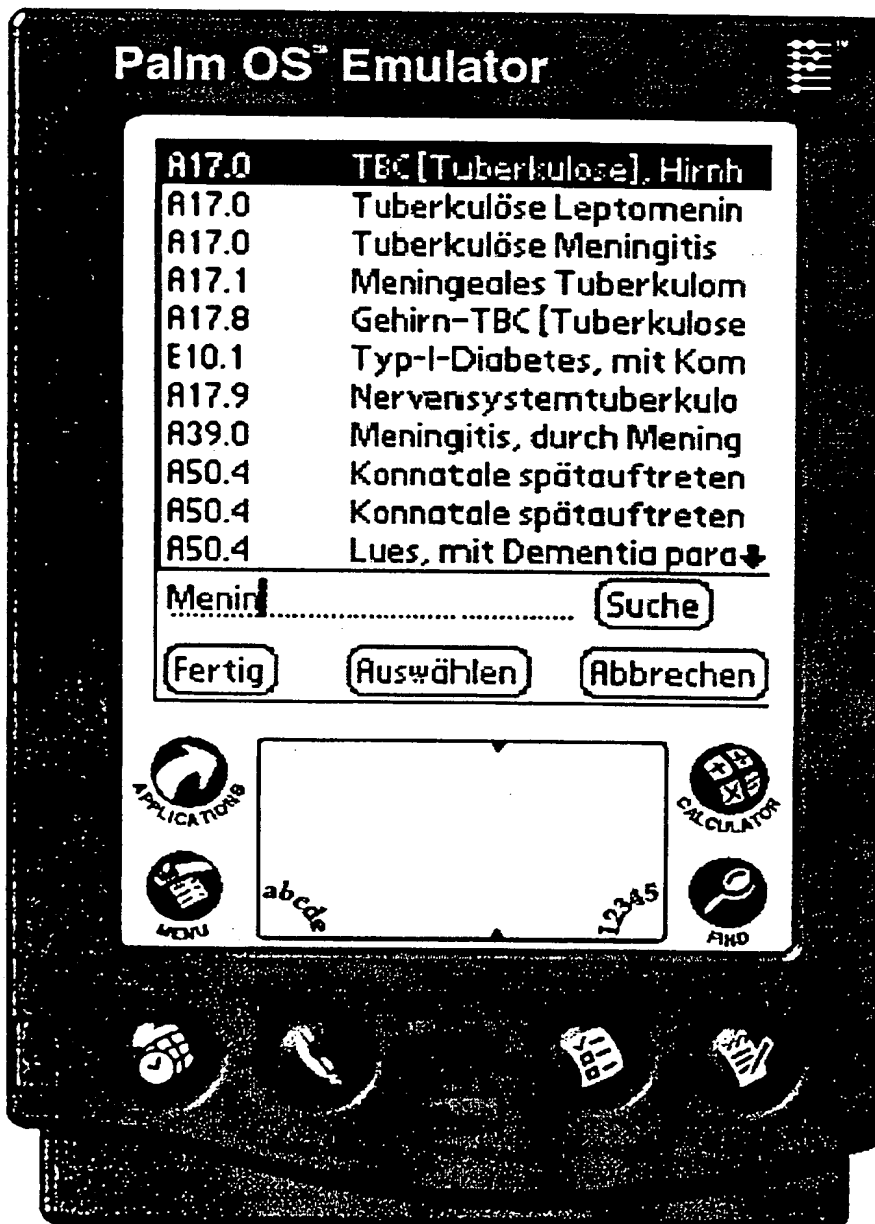
A17.0	TBC [Tuberkulose], Hirnh
A17.0	Tuberkulöse Leptomenin
A17.0	Tuberkulöse Meningitis
A17.1	Meningeales Tuberkulom
A17.8	Gehirn-TBC [Tuberkulose
E10.1	Typ-I-Diabetes, mit Kom
A17.9	Nervensystemtuberkulo
A39.0	Meningitis, durch Mening
A50.4	Konnatal spätauftreten
A50.4	Konnatal spätauftreten
A50.4	Lues, mit Dementia para↓

Menin

Suche

Fertig Auswählen Abbrechen

Figur 10



Diagnosen

A17.0	Tuberkulöse Meningitis	↑
A17.0	Tuberkulöse Meningitis	
A17.1	Meningeales Tuberkulom	
A17.1	Meningeales Tuberkulom	
A17.1	Meningeales Tuberkulom	

Code: A17.1 (Suchen)

Text: Meningeales Tuberkulom

Art: ▼

(Fertig) (Löschen)

Figur 11



Sauerbruch **Medikamente**

Datum: 26.04.01 Zeit: 15:34:5 (ToDo)

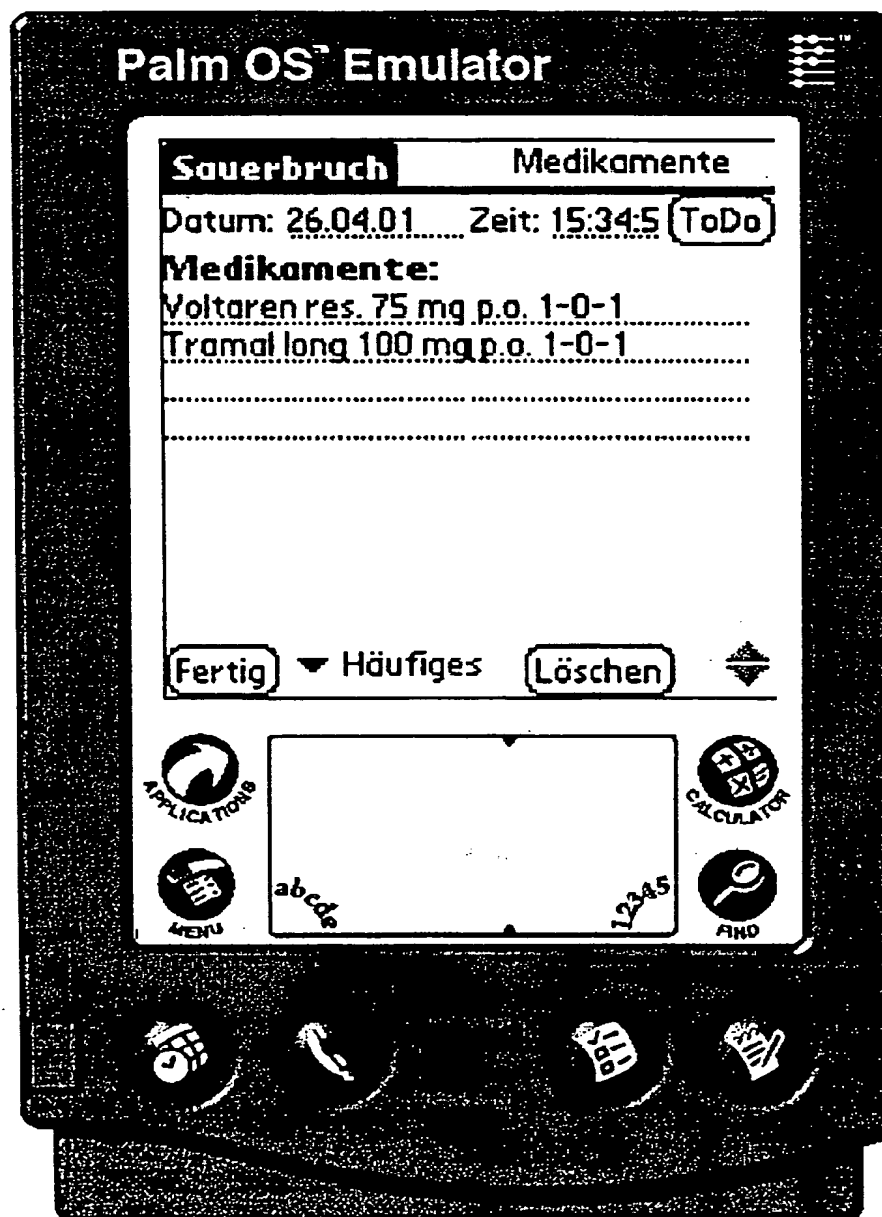
Medikamente:

Voltaren res. 75 mg p.o. 1-0-1

Framal long 100 mg p.o. 1-0-1

Fertig ▼ Häufiges Löschen

Figur 13



Sauerbruch	▼ Alle
Ferdinand	* 03.07.1875
26.04. Diag	Tuberkulöse Meningi
26.04. Med	Valtoren res. 75 mg p
Fertig	Drucken
▼ Neu	

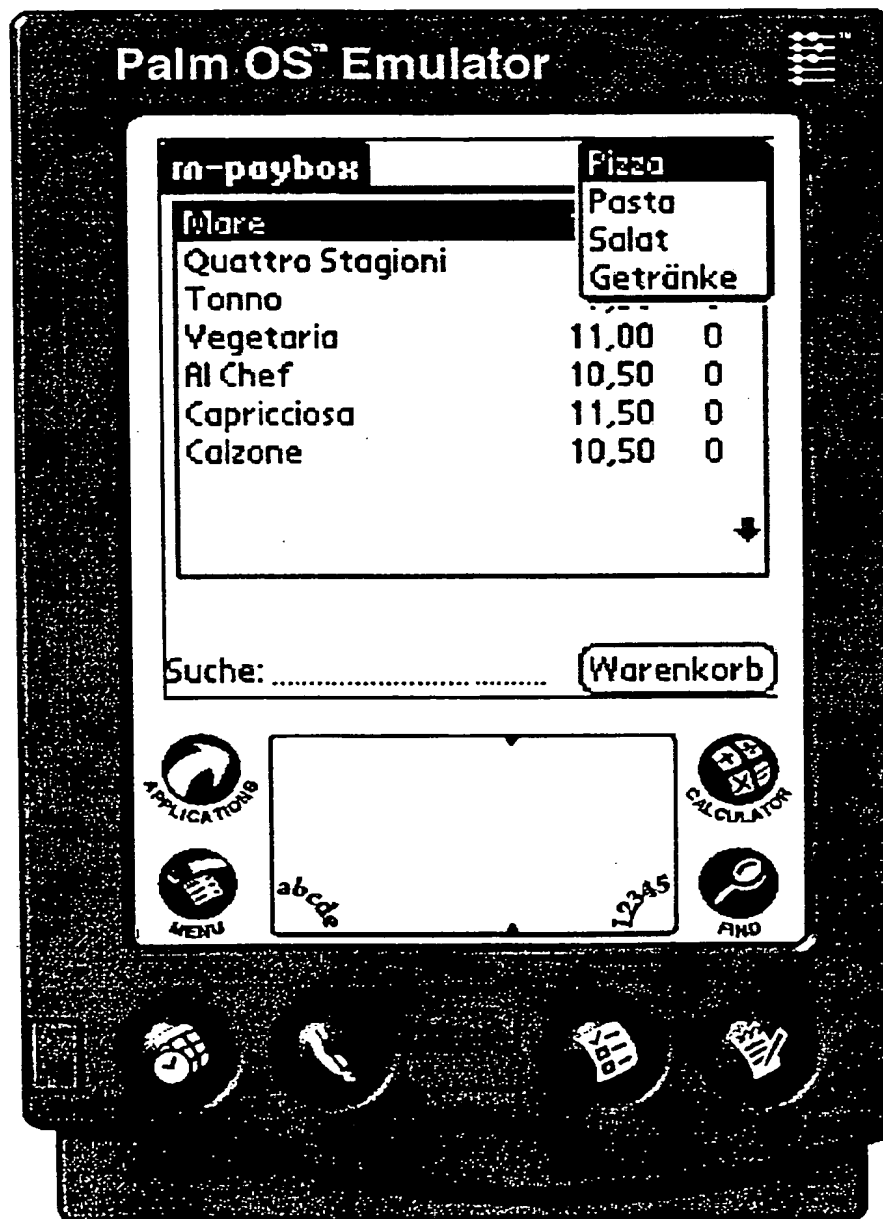
Figur 14



m-paybox		Pizza
Mare		Pasta
Quattro Stagioni		Salat
Tonno		Getränke
Vegetaria	11,00	0
Al Chef	10,50	0
Capricciosa	11,50	0
Calzone	10,50	0

Suche: Warenkorb

Figur 15



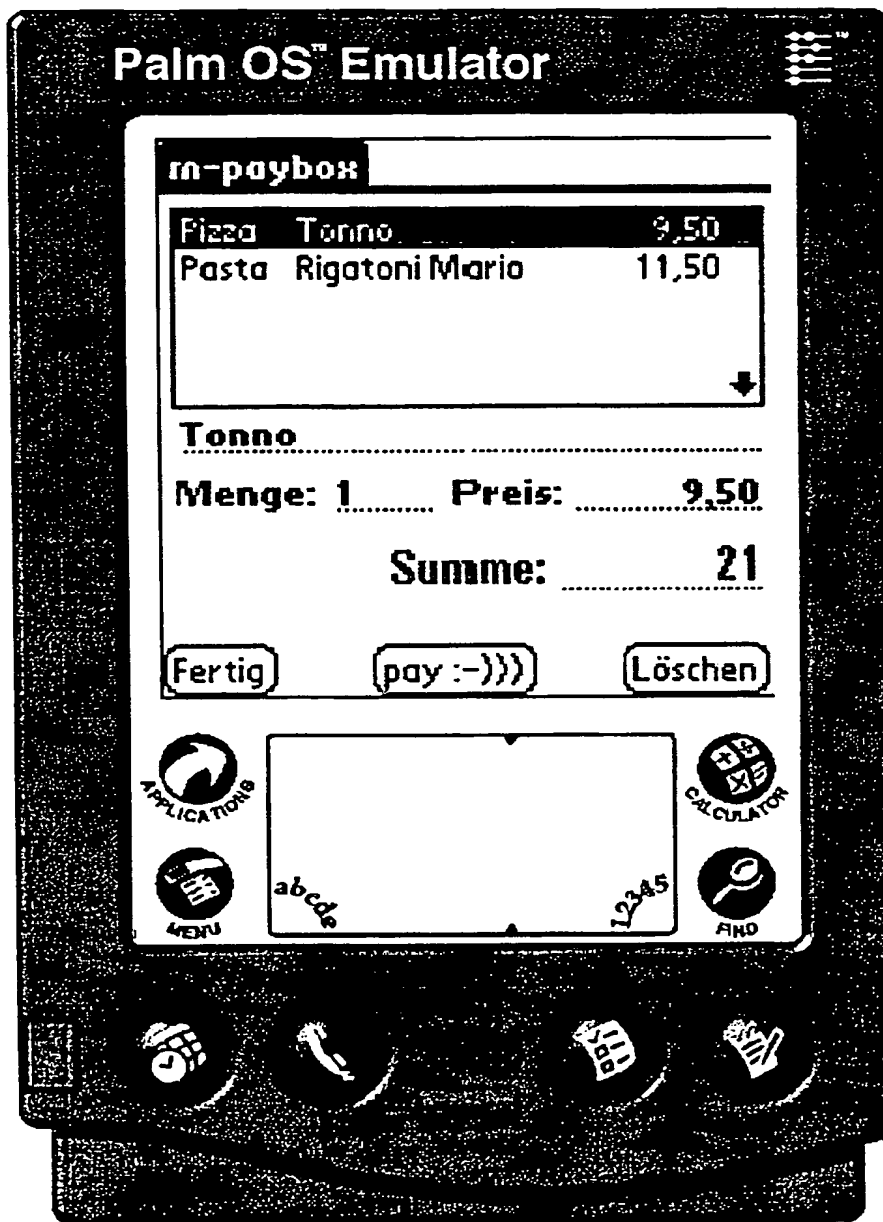
m-paybox	Pizza
Tonno	
Beschreibung:	
Thunfisch und Zwiebeln	
Menge: 1	
Fertig	Abbrechen

Figur 16



m-paybox		
Pizza	Tonno	9,50
Pasta	Rigatoni Mario	11,50
↓		
Tonno		
Menge: 1	Preis:	9,50
Summe:		21
Fertig	pay :->>>	Löschen

Figur 17



in-paybox

paybox: + 49

Betrag: DEM 21

1	2	3
4	5	6
7	8	9
0		

Erst die
Nummer der
paybox
eingeben und
dann "pay"!

Fertig pay :->>> Abbrechen

Figur 18

